**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**к курсовой работе**

**по дисциплине «Программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1323 |  | Чекстер А. А. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2022

**Цель работы.**

Объединить все 4 лабораторные работы в единый проект. Нужно добавить инфраструктуру переключения между заданиями (интерактивное меню).

**Основные теоретические положения.**

Вы делаете объявления в файле заголовка, а затем используете директиву #include в каждом CPP-файле или другом файле заголовка, который требует этого объявления. Директива #include вставляет копию файла заголовка непосредственно в CPP-файл перед компиляцией.

Затем создайте файл реализации (обычно с расширением CPP или аналогичного расширения). Мы вызовем файл my\_class.cpp и предоставим определение для объявления члена. Мы добавим директиву #include для файла "my\_class.h", чтобы объявление my\_class вставлено в этот момент в CPP-файле, и мы добавим <iostream> в объявление .std::cout Обратите внимание, что кавычки используются для файлов заголовков в том же каталоге, что и исходный файл, а угловые скобки используются для заголовков стандартной библиотеки. Кроме того, многие заголовки стандартной библиотеки не имеют расширения H или других файлов.

Теперь мы можем использовать my\_class в другом CPP-файле. Мы #include файл заголовка, чтобы компилятор извлекал объявление.

**Постановка задачи.**

Объединие всех 4 практических работ в единый проект.

**Выполнение работы.**

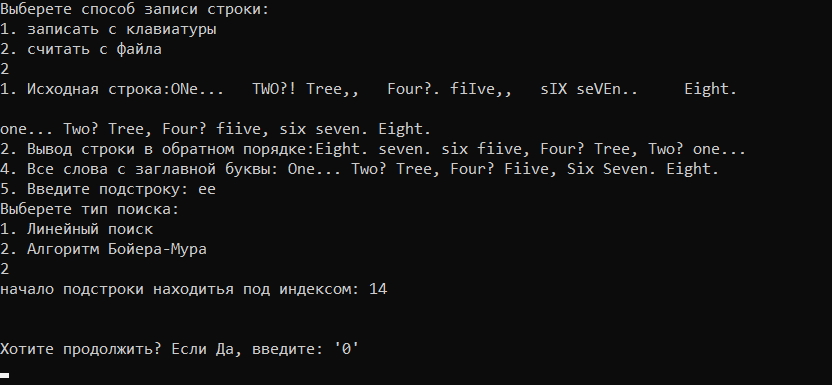
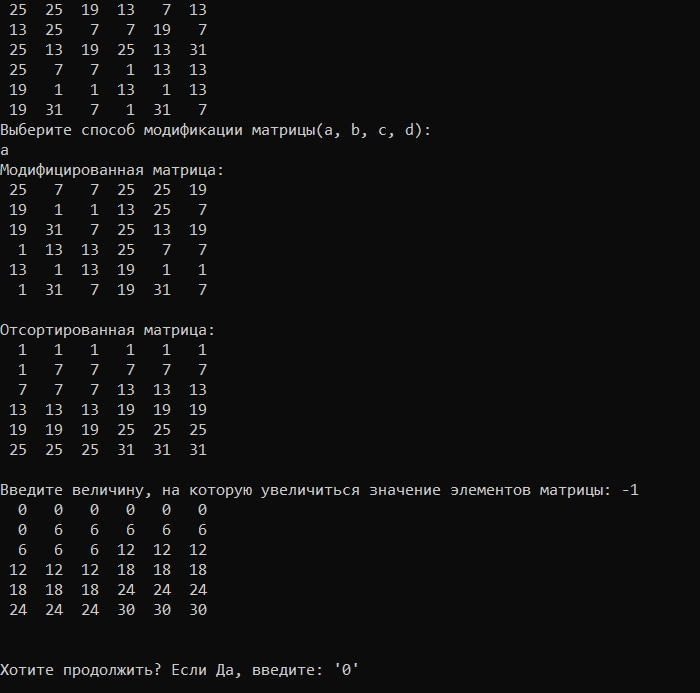
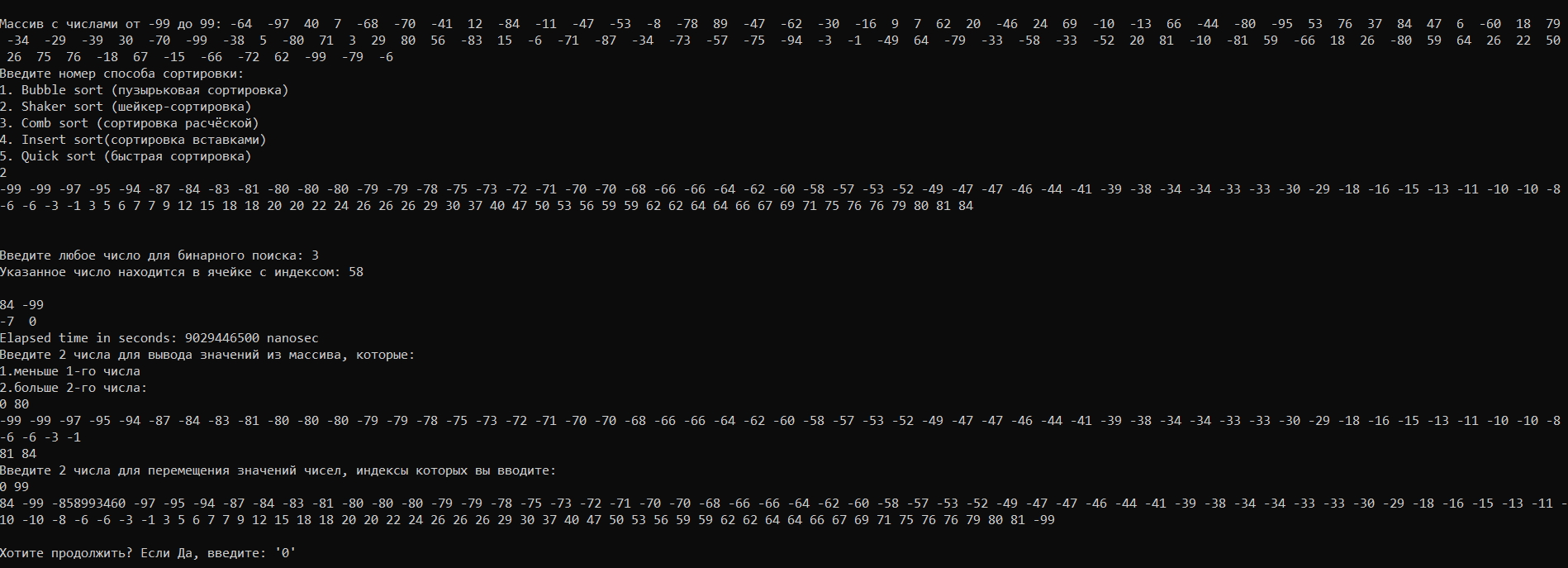
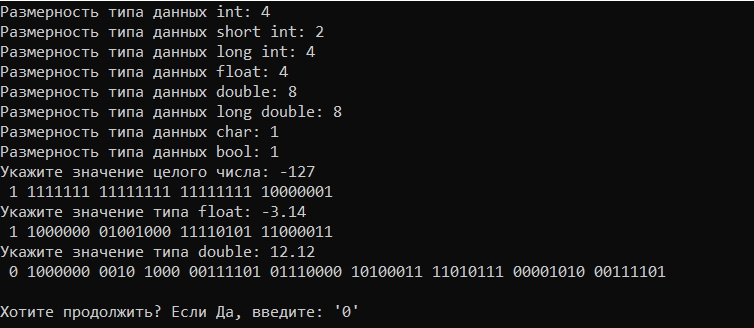
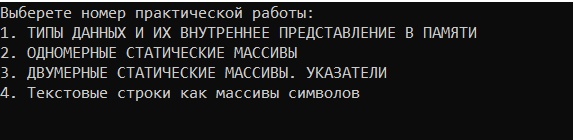
Часть кода программы представлена в приложении А.

Также код программы можно найти на сайте github по ссылке: https://github.com/Adelina2373/Coursovaya/tree/main

**Блок описания кода и использованных алгоритмов:**

1. Создаём (.cpp) файл курсовой работы. В нём реализовываем меню с возможностью выбора практической работы. В каждом пункте идёт обращение к каждому отдельному файлу.
2. Создаём общий (.h) файл и объявляем функции, которые в дальнейшем будем прописывать в отдельных файлах.
3. Создаём четыре файла (.cpp). В каждом реализовываем практические работы. Во всех (.cpp) файлах прописываем «#include "Practice.h"», чтобы связать файлы между собой.

**Блок скриншотов работы программы**



**Вывод.**

В процессе выполнения данной курсовой работы изучается объединение всех 4 практических работ в единый проект и добавление инфраструктуры переключения между заданиями, а также повторяется работа с файлами.

Приложение А

рабочий код

#include <iostream>

#include "Prctice.h"

using namespace std;

int main()

{

setlocale(0, "");

int snachala;

while (true) {

int vvod;

Menu: {

cout << "Выберете номер практической работы: \n"

"1. ТИПЫ ДАННЫХ И ИХ ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ В ПАМЯТИ\n"

"2. ОДНОМЕРНЫЕ СТАТИЧЕСКИЕ МАССИВЫ\n"

"3. ДВУМЕРНЫЕ СТАТИЧЕСКИЕ МАССИВЫ. УКАЗАТЕЛИ\n"

"4. Текстовые строки как массивы символов\n";

cin >> vvod;

}

switch (vvod) {

case 1: {

system("cls");

practice1();

system("pause");

}

case 2: {

system("cls");

practice2();

system("pause");

}

case 3: {

system("cls");

practice3();

system("pause");

}

case 4: {

system("cls");

practice4();

system("pause");

}

}

cout << "\n\nХотите продолжить? Если Да, введите: '0'" << endl;

cin >> snachala;

if (snachala != 0)

break;

}

}